

Reproducible probe-level analysis of the Affymetrix Exon 1.0 ST array with R/Bioconductor

Maria Rodrigo-Domingo, Rasmus Waagepetersen, Julie Støve Bødker, Steffen Falgreen, Malene Krag Kjeldsen, Hans Erik Johnsen, Karen Dybkær and Martin Bøgsted

Abstract

Alternative splicing is the post-transcriptional process by which a single gene can produce multiple transcripts and thereby protein isoforms. The presence of different transcripts of a gene across samples can be analysed by whole-transcriptome microarrays. Reproducing results from published microarray data represents a challenge due to the vast amounts of data and the large variety of pre-processing and filtering steps employed before the actual analysis is carried out. To ensure a firm basis for methodological development where results with new methods are compared with previous results it is crucial to ensure that all analyses are completely reproducible for other researchers.

We here give a detailed workflow on how to perform reproducible analysis of the GeneChip® Human Exon 1.0 ST Array at probe and probeset level solely in R/Bioconductor, choosing packages based on their simplicity of use. To exemplify the use of the proposed workflow we analyse differential splicing and differential gene expression in a publicly available dataset using various statistical methods.

We believe this study will provide other researchers with an easy way of accessing gene expression data at different annotation levels and with the sufficient details needed for developing their own tools for reproducible analysis of the GeneChip® Human Exon 1.0 ST Array.

Contact: maria@math.aau.dk

and results, 3) communication to other researchers and 4) comparison between different methods. Reproducible research is gaining relevance among the scientific community as shown by the number of papers published on the subject during the last years [2, 3, 4]. Ioannidis *et al.* showed that the results of only 2 out of 18 published microarray gene-expression analyses were completely reproducible [5]. This is why some authors demand that documentation and annotation, database accessions and URL links and even scripts with instructions are made publicly available [2]. Journals like Biostatistics have even appointed an Associate Editor for reproducible research, but still treat it as a “desirable goal” rather than a requirement [6]. Setting up a framework for reproducible research necessarily implies working with free and open-source software as for example R/Bioconductor [7, 8]. Additionally, using Sweave [9] (a tool for embedding R code in L^AT_EX documents [11]), enables automatic reports that can be updated with output from the analysis.

The main tool in this paper will be the Bioconductor package `aroma.affymetrix` [12] that can analyse all Affymetrix chip types with a (.CDF) Chip Definition File. The number of arrays (samples) that can be simultaneously analysed by `aroma.affymetrix` is virtually unlimited as the system requirements are just 1GB RAM, for any operating system [13]. This package is freely available and can easily be installed into R. The `aroma.affymetrix` website www.aroma-project.org is conceived as reference for all the possible microarrays that can be analysed with `aroma.affymetrix`, and does not focus specifically on the analysis of the GeneChip® Human Exon 1.0 ST Array (or exon array in short). Portable scripts for a fast and basic analysis can be obtained on request to `aroma.affymetrix`’s authors.

The analysis of exon array data in R/Bioconductor is not yet standard. There are several packages available, and it can be a tremendous effort for a newcomer to maneuver between them, and to overcome the numerous challenges associated with these packages. This paper aims to make this task easier and to provide a quick reference guide to

1 Introduction

In the field of microarrays it has traditionally been difficult to compare new methods to methods in published papers as the many methods available for pre-processing, summarizing and filtering make it almost impossible to work with the exact same data, even when the raw data is made available. That is why we consider reproducible research fundamental as it will facilitate easy 1) revision of papers, 2) access to data

`aroma.affymetrix`'s documentation. We also explain how to extract data for different statistical analyses and propose a method for gene annotation and for gene profile visualization. For this last step, we use the packages `biomaRt` and `GenomeGraphs` to annotate and visualize the transcripts in a genomic context.

The analysis workflow presented in this paper is carried out solely in R/Bioconductor, and the paper is available as a Sweave (`.Snw`) document that will allow the reader to reproduce our exact results. The `.Snw` document can also be converted into an R script and executed. The workflow starts by reading in the data, followed by background correction and quantile normalization. We then explain how to obtain transcript cluster -, probeset -, and probe-level estimates. Afterwards, different methods for the statistical analysis of differential splicing or differential gene expression are reviewed. Finally, we make a suggestion on how to annotate transcript clusters to genes in the lists obtained from the statistical analyses, and how to plot the data including genomic information. To exemplify the use of the workflow, an example dataset [14] is analysed along the way.

2 Background on alternative splicing

Splicing is the post-transcriptional process that generates mature eukaryotic mRNAs from pre-mRNAs by removing the non-coding intronic regions and joining together the exonic coding regions. For many genes, two or more splicing events take place during maturation of mRNA molecules resulting in a corresponding number of alternatively spliced mRNAs. These mature mRNAs translate into protein isoforms differing in their amino acid sequence and ultimately in their biochemical and biological properties [15, 16]. Alternative splicing is one of the main tools for generating RNA diversity, contributing to the diverse repertoire of transcripts and proteins [16, 17]. It is known that 92-94% of multi-exon human genes are alternatively spliced and that 85% of those have a minor isoform frequency of at least 15% [18, 19]. In our case we will focus on the detection of differential splicing between groups, as for instance tissue types, or healthy vs. diseased samples.

The exon array was presented in October 2005 as a tool for the analysis and profiling of whole-transcriptome expression [20, 21, 22]. To interrogate each potential exon with at least one probeset, the exon array contains about 5.6 million probes grouped into more than 1.4 million probesets (most probesets consisting of 4 probes), which are further grouped

into 1.1 million exon clusters, or collections of overlapping exons. Finally, exon clusters are grouped into over 300.000 transcript clusters to describe their relationship, as for example shared splice sites or overlapping exonic sequences. Each gene is covered, on average, by 40 probes interrogating regions located along the entire gene [23]. This probe positioning aims at providing better estimates of gene expression levels than previous arrays, and allows for the study of differential splicing [24] and differential gene expression based on summarized exon expression.

The exon array has three levels of annotation for the interrogated transcript clusters: *core*, *extended* and *full* [25]. Core transcript clusters are supported by the most reliable evidence such as RefSeq transcripts and full-length mRNAs [26] and a core transcript cluster is roughly a gene [27]; the extended level contains the core transcript clusters plus cDNA-based annotations [28] and the full level contains the two previous levels plus *ab-initio*, or algorithmic, gene predictions [29]. It is worth noting that `aroma.affymetrix` enables the analysis at the three levels of annotation mentioned above, and also that it provides intensity estimates for probes, probesets and transcript clusters, allowing for a variety of options for the analysis.

3 Workflow

Our workflow for the analysis of exon array data starts setting up the required folder structure for `aroma.affymetrix`. The data is then preprocessed and summarised at transcript cluster and/or probeset level. Next, transcript clusters are analysed with several statistical models to detect differential expression or splicing and the transcripts of interest are annotated and visualised at the end, see Figure 1. In the code, places where user input is needed are marked by `***`, and places where the user can choose whether to modify parameters are marked by `**`.

To exemplify the use of the tutorial we have used Affymetrix's colon cancer data set [14], consisting of a collection of paired samples of colon tumour tissue and adjacent normal tissue from 10 patients and available at http://www.affymetrix.com/support/technical/sample_data/exon_array_data.affx. According to Affymetrix's website, the RNA samples are from a commercial source. This dataset has been used in a number of papers to evaluate the performance of different analysis methods [30, 31, 32, 33] and a number of genes have been validated to present differential splicing or not [14]. The analysis was done in R version 2.15.1 (32 bit).

Start by installing and loading `aroma.affymetrix` in R and loading the other libraries required:

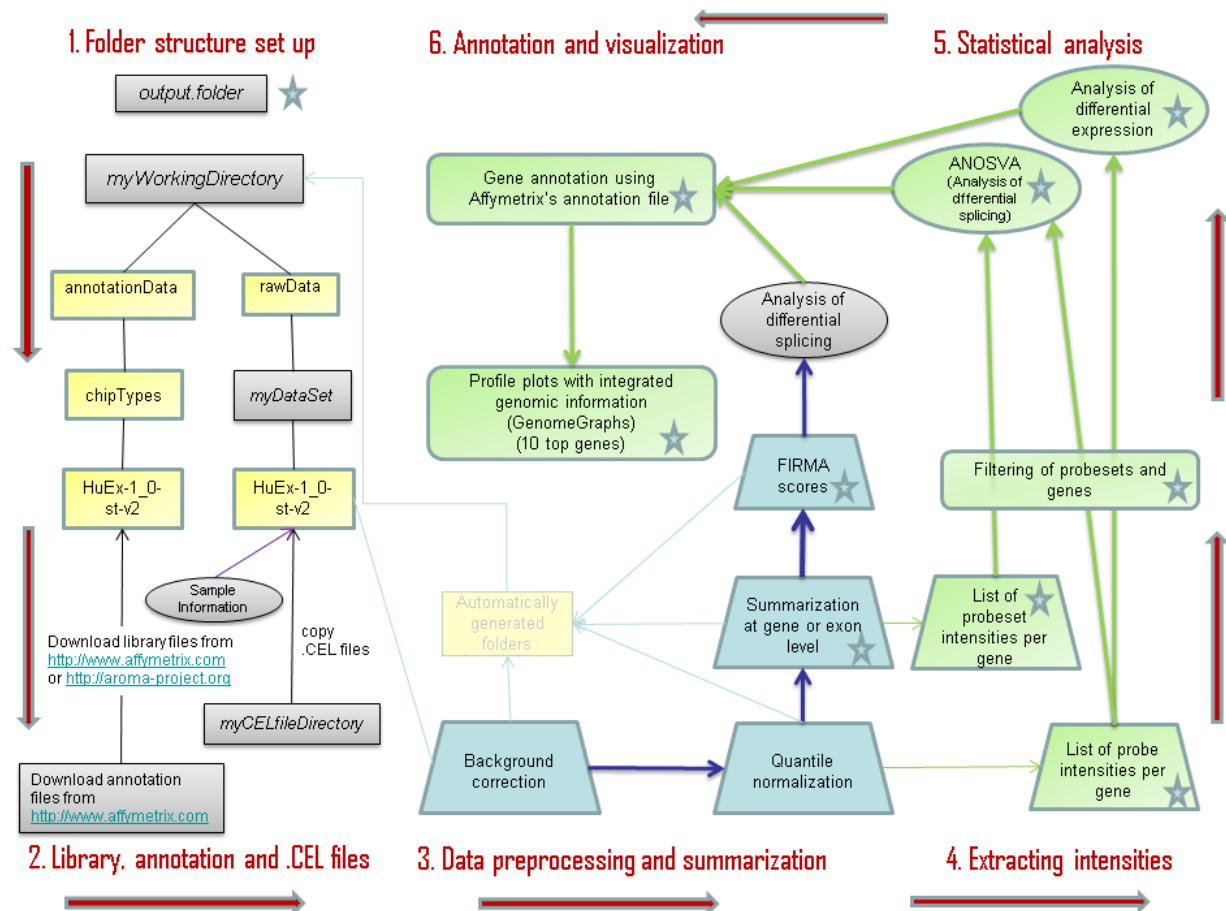


Figure 1: Flowchart for an analysis with `aroma.affymetrix`, read counter-clockwise starting in upper left corner: 1. and 2. folder structure set-up including library, annotation and .CEL files, 3. data pre-processing and summarization, 4. extraction of intensities at transcript cluster, probeset and probe level, including filtering recommended by Affymetrix, 5. statistical analysis of differentially expressed or spliced transcript clusters and 6. annotation and visualization of transcript cluster profiles. Blue boxes represent parts of the analysis implemented in `aroma.affymetrix`, yellow and green boxes are part of the code provided in this paper, and purple boxes represent user-input needed. Output produced at several steps is saved in user-chosen "output.folder" and represented by a star shape in the workflow. A number of folders are automatically generated by `aroma.affymetrix` - represented by a faded yellow rectangle in 3. -, our workflow does not make use of the contents of such folders.

```
> source("http://aroma-project.org/hbLite.R")
> hbInstall("aroma.affymetrix")
> require(aroma.affymetrix)
> require(biomaRt)
> require(GenomeGraphs)
```

3.1 Setting up the structure and files for the analysis workflow

This section corresponds to steps 1. and 2. in Figure 1. The first step is to create the folder structure: under a main folder of our choice - “myworkingDirectory” - we will create the “rawData” and “annotationData” folders, which will be common to all *aroma.affymetrix* projects. Inside “annotationData”, the subfolder “chipTypes” will contain one subfolder per chip type, with the exact name of the .CDF file provided by Affymetrix, “HuEx-1_0-st-v2” in our case. Inside this folder we will save any library and annotation files that might be needed. Besides, the “myDataSet” folder will be created under “rawData” to store .CEL files. These files are the output of a microarray experiment and contain the result of the intensity calculations per probe or pixel. Note that the microarray experiment produces one .CEL file per array and that one array analyses one sample. Affymetrix sometimes refer to their microarrays as “chips”. Note also that we need one “myDataSet” folder per experiment and that “myDataSet” will be added as a tag at the end of the *aroma.affymetrix* output.

```
> ##### user-defined working directory
> wd <- "myWorkingDirectory"
> ##### user-defined data set name
> ds <- "myDataSet"
```

In the second step we save our library (Chip Definition File (.CDF) in our case) and .CEL files in the corresponding folders. Affymetrix’s unsupported .CDF files can be downloaded from http://www.affymetrix.com/Auth/support/downloads/library_files/HuEx-1_0-st-v2.cdf.zip, note that registration is needed. For the exon array, Elizabeth Purdom has created a number of binary .CDF files based on Affymetrix’s text .CDF file [13] that are faster to query and more memory efficient. Such binary .CDF files for core, extended and full sets of probesets can be downloaded from <http://aroma-project.org/node/122>. In the example below we use the custom aroma file for core transcript clusters, which might be updated in the future. Our original .CEL files will be copied from the user-specified “myCELfileDirectory” into the exon “rawData” subfolder (the code is part of the .Snw version of this paper). The desired output folder specified in “output.folder” should exist in advance.

```
> *** download user-defined library file
> library.file <-
+   paste(annotation.data.exon,
+         "HuEx-1_0-st-v2,coreR3,A20071112,EP.cdf",
+         sep = "/")
> download.address <-
+   "http://bcgc.lbl.gov/cdfFiles/"
> file <- paste("HuEx-1_0-st-v2,A20071112,EP",
+             "HuEx-1_0-st-v2,coreR3,A20071112,EP.cdf",
+             sep = "/")
> custom.cdf <-
+   paste(download.address, file, sep = "")
> download.file(url = custom.cdf,
+             destfile = library.file,
+             mode = "wb", quiet = FALSE)
> **** user-defined directory containing .CEL files
> cel.directory <- "myCELfileDirectory"
> **** user-defined output folder
> output.folder <- "output.folder"
```

Besides, the sample information should be saved in a tab separated file with column names *celFile*, *replicate*, and *treatment* containing .CEL file name (without .CEL), replicate identifier and treatment name, respectively. This file should be called “SampleInformation.txt” and it will be copied from the user-specified directory into the “\rawData\myDataSet \HuEx-1_0-st-v2” folder (which will also contain the .CEL files) after it has been created. The sample information file for the colon cancer example is attached as an additional file.

```
> sample.info <-
+   read.table(file = paste(raw.data.exon,
+                           "SampleInformation.txt", sep = "/"),
+             sep = "\t", header = TRUE)
```

Finally, NetAffx transcript clusters’ and probesets’ annotation files should be saved in “annotationData/chipTypes/” “HuEx-1_0-st-v2”. We have used release 32, which was most up to date at the time of writing and we downloaded files “HuEx-1_0-st-v2.na32.hg19.transcript.csv.zip” and “HuEx-1_0-st-v2.na32.hg19.probeset.csv.zip” from http://www.affymetrix.com/estore/browse/products.jsp?productId=131452&categoryId=35676&productName=GeneChip-Human-Exon-ST-Array#1_3, *Technical Documentation* tab, under *NetAffx Annotation Files*. The extracted .csv files should be converted into .Rdata files for querying them faster in the future. Note that the number of lines to skip might differ for future annotation files.

```
> transcript.clusters.NetAffx.32 <-
+   read.csv(file = paste(annotation.data.exon,
+                           "HuEx-1_0-st-v2.na32.hg19.transcript.csv",
+                           sep = "/"), skip=24)
> probesets.NetAffx.32 <-
+   read.csv(file = paste(annotation.data.exon,
+                           "HuEx-1_0-st-v2.na32.hg19.probeset.csv",
+                           sep = "/"), skip=23)
```

3.2 Data pre-processing and summarization to probeset/transcript cluster level

After defining chip type and dataset, background correction and quantile normalization are carried out as shown in Figure 1, step 3. In these pre-processing steps, it is possible to use either Affymetrix's original .CDF file, the .CDF files provided by the aroma project (the file for core transcript clusters in our example), or a .CDF file created by the user. The summarization step, however, must be done using one of the custom .CDF files available at the `aroma.affymetrix` project website.

Background correction as defined by Irizarry [34] and quantile normalization are performed by the `RmaBackgroundCorrection()` and `QuantileNormalization()` functions, respectively. The raw, background corrected and quantile normalized probe intensities can be visualized using the `plotDensity()` function applied to the corresponding object. Summarization is done with the `ExonRmaPlm` function [35]. The parameter `mergeGroups` determines whether to summarize at transcript level (`TRUE`) or probeset level (`FALSE`). All the functions described automatically create subfolders such as "plmData" or "probeData" inside "myWorkingDirectory". A more detailed version of this code with interesting comments about the choice of .CDF and possibilities for quality control is available at <http://www.aroma-project.org/vignettes/FIRMA-HumanExonArrayAnalysis>.

```
> chipType <- "HuEx-1_0-st-v2"
> *** user-defined .CDF file: change tags parameter
> cdf <- AffymetrixCdfFile$byChipType(chipType,
+   tags = "coreR3,A20071112,EP")
> cs <- AffymetrixCelSet$byName(ds, cdf = cdf)
>
> # background correction
> bc <- RmaBackgroundCorrection(cs)
> csBC <- process(bc, verbose = verbose)
>
> # quantile normalization
> qn <- QuantileNormalization(csBC,
+   typesToUpdate = "pm")
> csN <- process(qn, verbose = verbose)
>
> # summarization
> # transcript cluster level
> plmTr <- ExonRmaPlm(csN, mergeGroups = TRUE,
+   tag = "coreProbesetsGeneExpression")
> # probeset/exon level
> plmEx <- ExonRmaPlm(csN, mergeGroups = FALSE,
+   tag = "coreProbesetsExonExpression")
```

3.3 Extraction of intensity estimates

The `aroma.affymetrix` documentation focuses on analyses at the probeset and transcript cluster levels.

The respective intensities are obtained by applying the function `getChipEffectSet()` to the transcript or probeset `plm` objects (`plmTr` and `plmEx`, respectively) and then extracting the corresponding dataframes. However, it is also possible to extract the background corrected and quantile normalized intensities of all probes using the function `getUnitIntensities`. While `plmTr` is suitable for the FIRMA analysis (3.4), `plmEx` is well suited for probeset-level analysis. For the linear model analysis described in Section 3.4 we have created one list of dataframes containing probe intensities per transcript cluster, and another list of dataframes containing probeset intensities per cluster (code included in `.Snw`).

```
> # extract a matrix of gene intensities
> cesTr <- getChipEffectSet(plmTr)
> trFit <- extractDataFrame(cesTr, addNames=TRUE)
> # extract a matrix of probeset intensities
> cesEx <- getChipEffectSet(plmEx)
> exFit <- extractDataFrame(cesEx, addNames=TRUE)
> # extract a list of probe intensities per gene
> unitIntensities <-
+   readUnits(csN, verbose=verbose)
```

The high number of transcript clusters analysed in combination with the usually small number of chips tends to cause a high number of false positives [25]. In order to reduce the number of false positives, Affymetrix recommends to perform detection above background (DABG) [36] on the dataset prior to the analysis [25]. The DABG procedure is not implemented in `aroma.affymetrix`, so we decided to follow the procedure described in [30] and use 3 as a threshold for the probeset intensity, so that probesets with a \log_2 intensity below 3 will be marked as absent. Except for this change, we followed the guidelines proposed in [25] to remove absent transcript clusters and probesets, where neither probesets that are absent in more than half of the samples of a group nor transcript clusters with more than half of the probesets absent are analysed.

Besides this filtering based on expression levels, another filtering that removes probesets presenting cross-hybridization is also advisable [25]. Cross-hybridising probesets are identified in file "HuEx-1_0-st-v2.na32.hg19.probeset.csv" and removed. Affymetrix recommends to filter them out after the analysis, but we have decided not to include them in the analysis in order to narrow down the number of probesets/transcript clusters to investigate.

The filtering procedure is part of the `.Snw` file. In our example, where we analysed only core probesets, 136233 probesets out of 284258 were deemed present by our filter, and the number of transcript clusters to analyse (present in both samples) was reduced from 18708 to 8401.

3.4 Statistical analysis

In this section we give an overview of model-based statistical methods available for the analysis of differential splicing and suggest a method for the analysis of differential gene expression.

3.4.1 Differential splicing

The analyses carried out in this paper are model-based approaches, and the analysis of differential splicing is done gene-wise. The models we use are extensions of the linear model by Li and Wong [37]

$$y_{pt} = \alpha_p + \beta_t + \varepsilon_{pt}, \quad (1)$$

where y_{pt} is the intensity measure of probe p for treatment t , α_p is a probe affinity term, β_t is the gene-level estimate for treatment t , and ε_{pt} is the error term.

ANOSVA (Analysis of Splicing Variation) was presented by Cline *et al.* [38] and is a two-way ANOVA model with probeset and treatment as factors:

$$y_{pet} = \mu + \alpha_e + \beta_t + \gamma_{et} + \varepsilon_{pet}, \quad (2)$$

where y_{pet} is the intensity measure of probe p in probeset e and treatment t , the overall mean μ is the baseline level of all probes in all experiments, and α_e and β_t are the probeset and treatment effects. The interaction term γ_{et} tells whether the effect of the probeset depends on the treatment and is therefore key to the detection of differential splicing.

The model in (2) can be extended to include random effects associated to replications from the same individual, r :

$$y_{petr} = \mu + \alpha_e + \beta_t + \gamma_{et} + I_r + C_{tr} + \varepsilon_{petr}, \quad (3)$$

where I_r is the random effect of each individual r and C_{tr} is the random chip effect. The error terms ε_{petr} are i.i.d. $N(0, \sigma^2)$ -distributed.

Under the null-hypothesis of no differential splicing, the γ_{et} 's will all be zero, therefore we consider the test statistic

$$t_{et} = \frac{\hat{\gamma}_{et}}{\hat{\sigma}},$$

where $\hat{\gamma}_{et}$ is the estimate for γ_{et} , and $\hat{\sigma}$ is the standard error of $\hat{\gamma}_{et}$. Large values will be critical for the null hypothesis. Under the model assumptions, t will follow a $t_{N-T \cdot (n_e + R - 1)}$ distribution [39, Chap. 5], with N the total number of observations per transcript cluster, T the number of treatments, n_e the number of probesets in the transcript cluster and R the number of individuals. For each gene, the smallest p -value from the above t -tests is regarded as a measure of confidence that the gene is differentially spliced across the experimental conditions [38]. The interaction estimates and variances and thus the test

statistics are contrast-dependent, so choosing a different contrast will alter the gene lists. In our analysis, we have used the **sum** contrast available in **R**, where parameter estimates are centred around zero.

A package in **R** for fitting linear and generalized linear mixed-effects models is **lme4** [52]. Linear mixed-effects models can also be analysed using **lm**, which is faster but requires a balanced design. In our example dataset we use **lm** to fit the model in equation (3) to the probe level estimates obtained using **unitIntensities()** from **aroma.affymetrix** to 8075 multiexonic transcript clusters. Here, we only show the code corresponding to equation (3), the rest of the code is part of the .Snw file.

```
> # ** user-defined parameters for linear model
> lm <-
+   lm(intensity ~ probeset + treatment +
+       C(probeset:treatment, contr.sum) +
+       replicate/treatment)
> n.probesets <- length(unique(dataframe$probeset))
> main.effects <-
+   1 + (n.probesets - 1) +
+   (length(unique(dataframe$treatment)) - 1)
> DS.parameters <- (n.probesets - 1)*
+   (length(unique(dataframe$treatment)) - 1)
> p.t <-
+   min(summary(lm)$coefficients[(main.effects+1):
+   (main.effects + DS.parameters), "Pr(>|t|)"])
```

Although the vast majority of probesets contain 4 probes, transcript clusters containing probesets with less than 4 probes will give rise to an unbalanced design. Nevertheless, the t -distribution is almost a normal distribution for long transcript clusters so the unbalanced design does not have any practical implications. For shorter transcript clusters, however, an unbalanced design might be a problem.

The top 10 most differentially spliced genes, sorted by the minimum p -value of their t -tests, appear in Table 1a. The gene *ZAK* was validated as differentially spliced in [14]. See Figure 2 for the profile plot of *KLK10*, where the thick lines representing the mean intensity in each group have been plotted for easing the interpretation. Note that there is one measurement per probe in each probeset, typically 4 probes per probeset. How to obtain such plots is described in Section 3.5 below. The genes in Figures 2, 3, 4 and 6 were chosen because they span over a shorter genomic region and show a more clear picture of the relationship between probesets and exons than the other genes in the lists of top 10 genes.

A slight variation of ANOSVA is the probeset-model as implemented in Partek [40] (note that the probe subscript p has been removed):

$$y_{etr} = \mu + \alpha_e + \beta_t + \gamma_{et} + I_r + C_{tr} + \varepsilon_{etr}. \quad (4)$$

For the probeset-level ANOSVA we used the probeset level estimates obtained by **affyPLM**(...

`mergeGroups = FALSE`). After filtering for non-present or cross-hybridising probesets, and absent transcript clusters, we were left with 9494 transcript clusters to study. These clusters are sorted according to the minimum p -value of the individual t -test scores for differential splicing, and the top 10 genes obtained appear in Table 1b. The genes *MMP11*, *ZAK* and *COMP* are in the top 10 genes for both the ANOSVA probe and the ANOSVA probeset models. Gene *TGFBI* appears in Figure 3.

FIRMA (Finding Isoforms using Robust Multi-chip Analysis) was first introduced by Purdom *et al.* [30] for the exon array. In presence of differential splicing the model in (1) will not fit and this will show up in the residuals. The linear model used is the following:

$$y_{petr} = \alpha_p + \beta_{tr} + \varepsilon_{petr}, \quad (5)$$

with α_p the probe affinity, β_{tr} the gene-level effect for chip tr and the error terms ε_{petr} are i.i.d. $N(0, \sigma^2)$ -distributed. Note that in contrast to the model in (3) we do not compute an overall gene-level estimate, but a gene-level estimate per chip.

As in model (1), there is no treatment/probeset interaction term, so differential splicing is analysed probesetwise, using the residuals per probeset e :

$$r_{petr} = y_{petr} - (\hat{\alpha}_p + \hat{\beta}_{tr}), \quad (6)$$

$$p = 1, \dots, n_e, t = 1, \dots, T, r = 1, \dots, R$$

where $\hat{\alpha}_p$ and $\hat{\beta}_{tr}$ are the estimates of α_p and β_{tr} .

The median of the standardized residuals per probeset per chip is chosen as score statistic:

$$F_{e(tr)} = \text{median}_{p=1, \dots, n_e} \left(\frac{r_{petr}}{\text{MAD}} \right). \quad (7)$$

The standardization with the median absolute deviation (MAD) of the residuals per gene makes the scores comparable across transcript clusters.

The FIRMA scores are extracted from the `plmTr` object obtained in Section 3.2. All probesets and transcript clusters were analysed by FIRMA, since it is part of the default `aroma.affymetrix` workflow.

```
> firma <- FirmaModel(plmTr)
> fit(firma, verbose = verbose)
> fs <- getFirmaScores(firma)
> firma.scores <- extractDataFrame(fs)
```

After obtaining the FIRMA scores per probeset per sample, we proceeded as described in [30]: 1) For each probeset we took the difference of FIRMA scores for each of the 10 pairs of normal/cancer samples and 2) calculated the mean of the 10 differences per probeset. Then 3) we ranked the probesets according to their absolute mean difference: as the scores are comparable across transcript clusters, larger average differences between the normal and cancer samples will

point at exons more differentially spliced between the two conditions. The resulting list was filtered to keep only probesets and transcript clusters that has passed the filter described in 3.3 above, the code is part of the `.Snw` file. In order to get a gene list instead of a probeset list as in [30] we mapped probesets to transcript clusters and then selected the top 10 genes on the list, see Table 1c. The profile plot of gene *LGALS4* appears in Figure 4. The very high average difference for this gene is due to a FIRMA score of 830030.8 at probeset *3861578* corresponding to the tumour sample of replicate 7.

Out of 14 genes investigated for differential splicing in [14, Table 1], 10 passed our filtering procedure and were analysed for differential splicing: *ACTN1*, *VCL*, *CALD1*, *SLC3A2*, *COL6A3*, *CTTN*, *FN1*, *MAST2*, *ZAK* and *FXYD6*. The gene *ZAK* appears in the top 10 genes from ANOSVA probe and ANOSVA probeset, but a plot is not produced automatically by the code because *ZAK* is not recognised by BioMart. Instead, we looked the gene up in PubMed obtaining the Ensembl ID: ENSG00000091436, and used this to plot the gene, see Figure 5. The gene *COL6A3* appears among the top 100 genes for the ANOSVA probe and the ANOSVA probeset methods. Gene *ACTN1* is among the top 100 genes for ANOSVA probeset and it is also the first of Gardina's genes to appear in the filtered FIRMA list, at position 154.

3.4.2 Differential gene expression

In this section we analyse differential gene expression using probe level data. We study two types of transcript clusters: 1) the ones not included in the ANOSVA probe analysis above (with only one probeset or not present in both normal and cancer groups) and 2) the ones not showing differential splicing (ANOSVA probe p -value above 0.1). The analysis of group 2) is based on the hierarchical principle: only look for significant main effects (differential expression in this case) among those transcript clusters with no significant interaction terms (differential splicing) [46, p. 427]. In total, we analysed 16231 transcript clusters. We fit the following linear model to those transcript clusters:

$$y_{petr} = \mu + \alpha_e + \beta_t + I_r + C_{tr} + \varepsilon_{petr}, \quad (8)$$

where β_t is the gene level treatment effect, α_e is a parameter that captures probesets expressed above or below the overall transcript cluster level and I_r and C_{tr} are random effects for patient and chip, respectively.

```
> aov <-
+   aov(intensity ~ probeset + treatment +
+       Error(replicate + replicate:treatment))
```

Gene symbol	min p -value	Gene symbol	min p -value
MMP11	$2.98e-10$	SOX4	$9.07e-15$
SOX9	$2.59e-09$	MMP11	$9.48e-14$
FOXQ1	$6.38e-09$	ZAK	$4.40e-13$
KLK10	$3.84e-08$	FOXQ1	$5.98e-13$
SYNM	$4.53e-07$	TGFBI	$4.83e-12$
PHLDA1	$4.85e-07$	UBAP2L	$8.17e-12$
ZAK	$4.89e-07$	COMP	$2.87e-11$
SNTB1	$5.33e-07$	SLC2A1	$4.16e-11$
COMP	$6.85e-07$	CDH11	$4.19e-10$
XPOT	$7.87e-07$	CPXM1	$4.47e-10$

(a) Model: ANOSVA probe. (b) Model: ANOSVA probeset.

Gene symbol	Average difference
LGALS4	83000.0
HMGCS2	82.9
RPS6KA1	31.8
MUC13	28.2
RPL35	24.5
RPL35A	18.8
SLC39A14	18.2
TGFBI	16.2
COL23A1	15.5
SFT2D2	14.2

(c) Model: FIRMA.

Table 1: Top 10 differentially spliced genes from the models described in 3.4.1. Tables (a) and (b) show genes with minimum p -values, table (c) shows genes with their top scores. Genes highlighted in gray appear in Figures 2, 3, 4 and 5.

```
> p.F <-
+ summary(aov)$"Error: Within"[[1]]["treatment",
+ "Pr(>F)"]
```

The null hypothesis is that the gene expression is the same in all groups. The top 10 genes, with adjusted p -values appear in Table 2. The method used for adjusting the p -values was Benjamini-Hochberg's correction [47] using the function `p.adjust(..., method = "BH")`. Only 80 out of 159 genes appearing in Gardina's list of genes up- and down-regulated in tumour [14, Additional file 1] passed our filters and were analysed for differential gene expression. Among those analysed, the genes *CLDN1*, *SST*, *MUSK*, *KIAA1199* and *SLC30A10* were in the top 100. The gene *BEST4* is shown in Figure 6. This gene is down-regulated in tumour (blue) compared to normal (red) samples. The thicker lines represent the mean expression levels in the two groups.

3.5 Gene annotation and visualization

We chose to annotate transcript clusters to genes using the NetAffx transcript cluster annotation release 32 specified in Section 3.1 using the `AnnotateGenes()` function. Some transcript clusters present unspecific annotation and have several pos-

Gene symbol	p -value
LARGE	0.00287
IL6R	0.00847
RXRG	0.00847
BAI3	0.00968
C1orf175	0.01090
GRIK3	0.01090
BEST4	0.01090
KCNN3	0.01090
PLEKHH2	0.01090
HOXD10	0.01090

Table 2: Top 10 differentially expressed genes from (8) with corrected p -values. Gene *BEST4*, highlighted in gray, appears in Figure 6.

sible associated gene names. We have decided to remove such clusters from our output as we cannot map them uniquely to a gene and afterwards interpret the result according to the gene structure.

After gene annotation, the user can select genes for visual inspection. Visual inspection of candidates for differential splicing is recommended by Affymetrix as a way to identify possible false positives [25]. Plots of gene profiles with integrated genomic information are obtained using the `biomaRt` [48] and `GenomeGraphs` [49] packages in Bioconductor.

We use `bioMart` to connect to the latest version of the Homo Sapiens dataset in Ensembl [50] (Ensembl genes 68, GRCh37.p8 at the time of writing) and retrieve genes by their HGNC symbol [51]. Gene and exon structures are imported from Ensembl. `Gene` and `Exon` objects are created by `makeGene()` and `makeTranscript()` from `GenomeGraphs`. We store expression data and probeset start and stop positions in an `ExonArray` object by `makeExonArray()`. The final plot is created passing a list with the objects created to the `gdPlot(list(exon, gene, transcript,...))` function.

Our plots show on top the gene HGNC symbol followed by (+) for genes on the forward strand and by (-) for those on the reverse strand. Below, the plot of probeset intensities appears with vertical lines delimiting probesets. Note that for models based on probe-level data (ANOSVA probe, FIRMA and differential expression), the intensities of all probes in the probeset (1 to 4) are shown. Samples from the same treatment group appear in the same colour, red for normal samples and in blue for tumour samples in this case. Immediately after the profile plot, the gene model retrieved from Ensembl is shown in orange, followed by the possible transcript model(s) in blue. The gene model consists of the exons that appear in all possible transcript models. Exons (boxes) in the

gene model are linked by blue lines to the probesets above, indicating which probeset(s) interrogate which exon. See Figures 2 to 6.

4 Discussion

The aim of this paper was to give a tutorial on how to perform a complete and reproducible analysis of exon array data in R/Bioconductor. We have worked with three packages: `aroma.affymetrix`, `biomaRt` and `GenomeGraphs` to go from .CEL files to intensity data, statistical analysis, annotation and visualization. The packages were chosen for their flexibility and easy of integration. We believe that our workflow covers a number of analysis variants for the exon array, including differential splicing analysis at probe and probeset-level and differential expression analysis at probe level, and gives the user the opportunity to focus on all or only some of the aspects of the data analysis. We make our entire code available so that other researchers can use it as it is or adapt it to their needs.

Different Bioconductor packages could have been used in some of the analysis steps. For example, `xmapcore` [54] provides annotation data and cross-mappings between genetic features as transcript clusters or exons and Affymetrix probesets. This package, however, requires the separate installation of a MySQL database, which makes this a more complex alternative than the one we have chosen. The `xps` package [55] could have been used for data preprocessing and summarization, but it requires the installation of the R00T framework [56], and a certain level of understanding of R00T files and R00T trees is recommended. Our workflow does not require any prior knowledge beyond R/Bioconductor. Other free software includes BRB-Array Tools [57], based on R, C, Fortran and Java, with an Excel front end, and dChip [58], which is written in Visual C++ and developed for Windows - though some users have been able to run it on Mac and Unix computers.

A previous paper on exon arrays [24] suggests a pragmatic approach and does the analysis piecewise starting with Affymetrix Power Tools (APT) and then exporting the data to R. We recognise this is a fix for the lack of straightforward packages for dealing with the exon array in Bioconductor. However, it implies working with several pieces of software so we do not find it fit for reproducible research. Licensed software, like Partek [40] or GeneSpring GX [59], has been used in other studies [60, 22]. In contrast to licensed software, R/Bioconductor is free and available for anyone, it allows the user to control most analysis options and it enables customizable and reproducible analyses that are more easily re-

viewed. Still, the `aroma.affymetrix` package does not provide the speed of Affymetrix Power Tools or the licensed software, and it requires more user input. Nevertheless, with this code and minimal user input, any dataset can be analysed regarding differential expression at probe level and differential splicing using the ANOSVA model.

Differential gene expression could have been analysed using the gene level estimates obtained from the `plmTr` object in other R packages such as `limma` [53]. Another extension of the workflow could include a general analysis strategy of the FIRMA scores, which in this study was tailor-made for a two-treatment scenario. Besides, the profile plots we obtained with `GenomeGraphs`, though highly informative, are difficult to interpret for genes spanning over a long genomic region, as for example *TGFBI* in Figure 3 and *ZAK* in Figure 5. In the future, it would be interesting to study the flexibility of the output imported from Ensembl and, for example, remove the intronic regions from the gene and transcript models.

Acknowledgements/Author declaration

The authors would like to thank Henrik Bengtsson for helpful answers and comments regarding `aroma.affymetrix`. The authors would also like to thank Hang Phan and Anders Ellern Bilgrau for helpful discussions and ideas about code implementation and data plotting.

MRD designed and implemented the workflow, analysed data and drafted the manuscript. RW assisted in paper drafting and statistical modelling and interpretation. JSB, SF and MKK tested the workflow. JSB, MKK, HEJ and KDS assisted with data interpretation and participated in paper drafting. MB conceived and tested the workflow, assisted in manuscript drafting and coordinated the work. All authors read and approved the final manuscript.

References

- [1] Hothorn T and Leisch F. Case studies in reproducibility. *Briefings in Bioinformatics*, 2011.
- [2] Baggerly KA. Disclose data in all publications. *Nature*, 2010, 467. 149-155.
- [3] Baggerly KA and Coombes KR. Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics*, 2009, 3:1309-1334.
- [4] Coombes KR, Wang J, and Baggerly KA. Microarrays: retracing steps. *Nature Medicine*, 2007, 13. Correspondence.

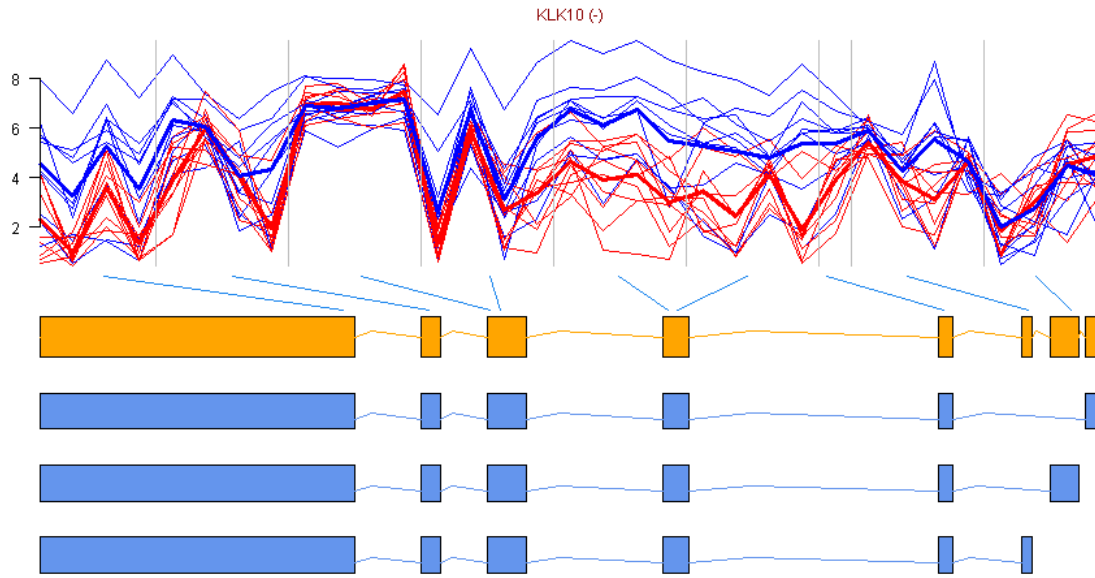


Figure 2: Profile plot of gene *KLK10* with the gene model and possible transcripts retrieved from Ensembl [50]. The (-) next to the gene name indicates that it is on the reverse strand. The mean intensities of each group are plotted with a thicker line. Exons 5 (mapped by probesets 4 and 5) and 8 (mapped by probeset 9), counted from the 5' end, seem to be higher expressed in tumour samples (blue) than in normal (red).

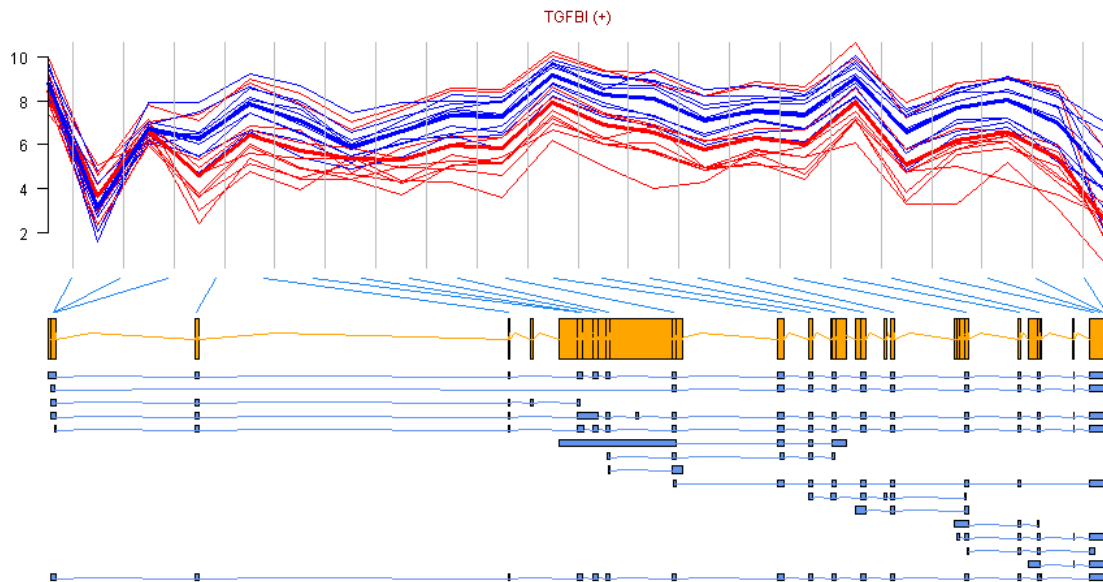


Figure 3: Profile plot of *TGFBI* with the gene model and transcripts retrieved from Ensembl [50]. The gene is on the forward strand as indicated by (+) next to its name. The mean intensities of each group are plotted with a thicker line, note that only one estimate is plotted by probeset, it corresponds to the estimate computed by `ExonRmaPlm(..., mergeGroups=FALSE)`. Here, it seems like the tumour samples (blue) present increased expression from exon 3 until the end of the transcript, with respect to normal (red) samples. Given that *TGFBI* is on the forward strand and that the difference is at the beginning of the transcript, we might be observing a case of alternative promoter usage.

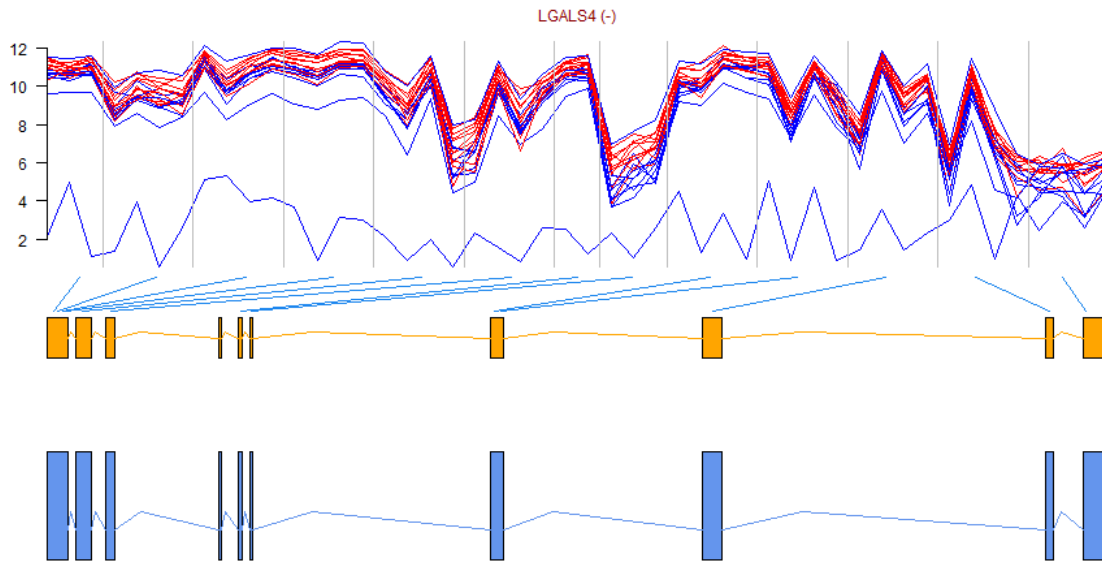


Figure 4: Profile plot of gene *LGALS4* with the gene model and transcripts retrieved from Ensembl [50], the gene is on the reverse strand. Interestingly, this gene only has one transcript. We might then be facing a novel splicing event or, more likely, a false positive detected by the method.

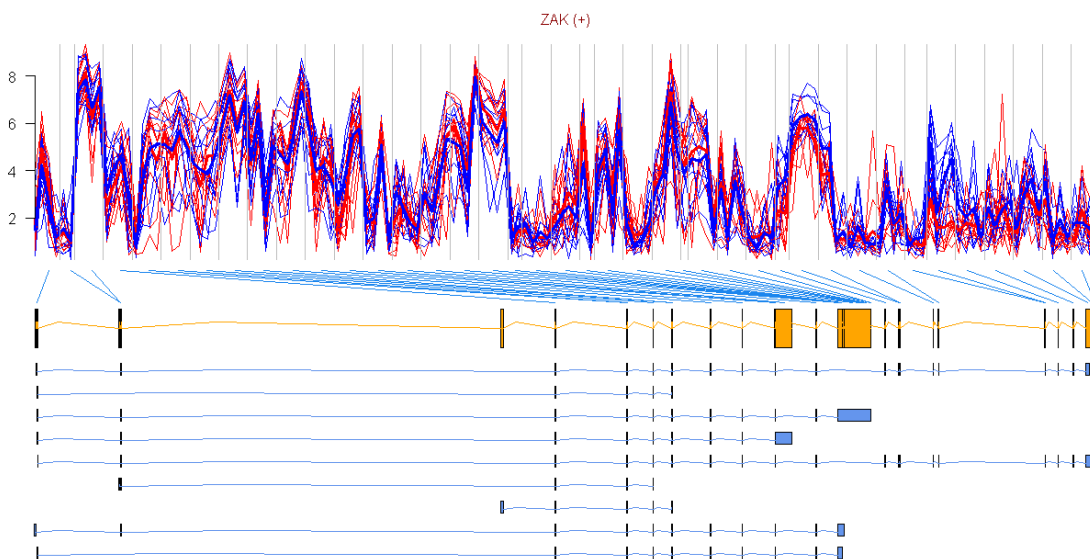


Figure 5: Profile plot of gene *ZAK* with the gene model and transcripts retrieved from Ensembl [50]. There seems to be a differential splicing event identified by probesets 5 and 6 (counted from the 3' end of the gene), corresponding to exons 3 and 4 from the 3' end.

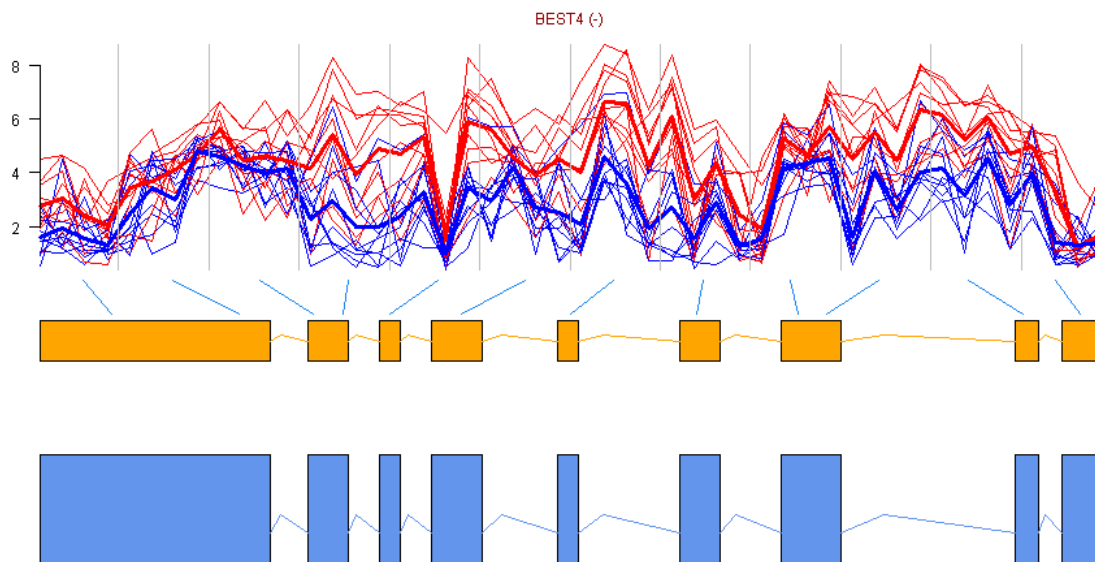


Figure 6: Profile plot of gene *BEST4*, which is on the reverse strand, with the gene model and transcripts retrieved from Ensembl [50]. The gene is under expressed in tumour (blue) compared to normal (red) samples. The thin lines are the expression levels for each sample, while the thicker lines represent the mean intensities in each of the two groups.

- [5] Ioannidis JP, Allison DB, Ball CA, et al. Repeatability of published microarray gene expression analyses. *Nature Genetics*, 2009, 41:149–155.
- [6] Peng RD. Reproducible research and *Biostatistics*. *Biostatistics*, 2009, 10:405–408.
- [7] R Development Core Team . *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011.
- [8] Gentleman RC, Carey VJ, Bates DM, et al. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 2004, 5:R80.
- [9] Leisch F. Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In Härdle Wolfgang and Rönz Bernd, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002.
- [10] Ramsey N. *Noweb home page*, November 2012. <http://www.cs.tufts.edu/~nr/noweb/>.
- [11] Lamport L. *L^AT_EX: A Document Preparation System*. Addison-Wesley Pub. Co, 2 edition, 1994.
- [12] Bengtsson H, Simpson K, Bullard J, et al. aroma.affymetrix: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory. Technical Report 745, Department of Statistics, University of California, Berkeley, 2008.
- [13] Bengtsson H, Bullard J, Hansen K, et al. *Aroma project*, January 2012. <http://www.aroma-project.org/>.
- [14] Gardina PJ, Clark TA, Shimada B, et al. Alternative splicing and differential gene expression in colon cancer detected by a whole genome exon array. *BMC Genomics*, 2006, 7.
- [15] Black DL. Mechanisms of alternative pre-messenger RNA splicing. *Annual Reviews of Biochemistry*, 2003, 72:291–336.
- [16] Hallegger M, Llorian M, and Smith CWJ. Alternative splicing: global insights. *The FEBS journal*, 2010, 277(4):856–866.
- [17] Licatalosi DD and Darnell RB. RNA processing and its regulation: global insights into biological networks. *Nature Reviews Genetics*, 2010, 11(1):75–87.
- [18] Pan Q, Shai O, Lee LJ, et al. Deep surveying of alternative splicing complexity in the hu-

- man transcriptome by high-throughput sequencing. *Nature Genetics*, 2008, 40:1413–1415.
- [19] Wang ET, Sandberg R, Luo S, et al. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 2008, 456:470–476.
- [20] Clark TA, Schweitzer AC, Chen TX, et al. Discovery of tissue-specific exons using comprehensive human exon microarrays. *Genome Biology*, 2007, 8.
- [21] Suzuki H, Osaki K, Sano K, et al. Comprehensive analysis of alternative splicing and functionality in neuronal differentiation of P19 cells. *PLoS ONE*, 2011, 6.
- [22] Thorsen K, Schepeler T, Oster B, et al. Tumor-specific usage of alternative transcription start sites in colorectal cancer identified by genome-wide exon array analysis. *BMC Genomics*, 2011, 12.
- [23] Affymetrix . Application focus: Whole-transcript Expression Analysis. Gene expression. Technical Report 702503-2, Affymetrix Inc., Santa Clara, CA, 2007.
- [24] Lockstone HE. Exon array data analysis using Affymetrix power tools and R statistical software. *Briefings in Bioinformatics*, 2011.
- [25] Affymetrix . Technical note: Identifying and Validating Alternative Splicing Events. An introduction to managing data provided by GeneChip® Exon Arrays. Technical Report 702422, Affymetrix Inc., Santa Clara, CA, 2006.
- [26] Pruitt KD, Tatusova T, and Maglott DR. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 2007, 33.
- [27] Affymetrix . Technical note: GeneChip® Exon Array Design. Technical Report 702026, Affymetrix Inc., Santa Clara, CA, 2005.
- [28] Benson DA, Karsch-Mizrachi I, Clark K, et al. GenBank. *Nucleic Acids Research*, 2012, 40.
- [29] Burge C and Karlin S. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 1997, 268:78–94.
- [30] Purdom E, Simpson KM, Robinson MD, et al. FIRMA: A method for detection of alternative splicing from exon array data. *Bioinformatics*, 2008, 24:1707–14.
- [31] Zheng H, Hang X, Zhu J, et al. REMAS: a new regression model to identify alternative splicing events from exon array data. *BMC Bioinformatics*, 2009, 10 Suppl 1.
- [32] Affymetrix . Alternative Transcript Analysis Methods for Exon Arrays. *Affymetrix GeneChip® Exon Array Whitepaper Collection*, 2005.
- [33] Turro E, Lewin A, Rose A, et al. MMBGX: a method for estimating expression at the isoform level and detecting differential splicing using whole-transcript Affymetrix arrays. *Nucleic Acids Research*, 2010, 38(1).
- [34] Irizarry RA, Hobbs B, Collin F, et al. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Bio-statistics (Oxford, England)*, 2003, 4(2):249–64.
- [35] Irizarry RA. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research*, 2003, 31(4).
- [36] Affymetrix . Statistical algorithms reference guide. Technical Report 701110, Affymetrix Inc., Santa Clara, CA.
- [37] Li C and Wong WH. Model-based analysis of oligonucleotide arrays: expression index computation and outlier analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 2001, 98:31–36.
- [38] Cline MS, Blume J, Cawley S, et al. ANOSVA: A statistical method for detecting splice variation from expression data. *Bioinformatics*, 2005, 21:i107–15.
- [39] Chambers JM and Hastie TJ. *Statistical Models in S*. Wadsworth & Brooks/Cole, 1992.
- [40] Inc. Partek. *Partek® Genomics Suite™*, 2008. Revision 6.3.
- [41] Rasche A and Herwig R. ARH: predicting splice variants from genome-wide data with modified entropy. *Bioinformatics*, 2010, (26):84–90.
- [42] Abdueva D, Wing MR, Schaub B, et al. Experimental comparison and evaluation of the Affymetrix exon and U133Plus2 GeneChip arrays. *PLoS ONE*, 2007, (2):e913.
- [43] Della Beffa C, Cordero F, and Calogero R. Dissecting an alternative splicing analysis workflow for GeneChip® Exon 1.0 ST Affymetrix arrays. *BMC Genomics*, 2008, 9.

-
- [44] Cuperlovic-Culf M, Belacil N, Culf A, et al. Data analysis of alternative splicing microarrays. *Drug Discovery Today*, 2006, (11):983–990.
- [45] Laajala E, Aittokallio T, Lahesmaa R, et al. Probe-level estimation improves the detection of differential splicing in affymetrix exon array studies. *Genome Biology*, 2009, (10).
- [46] Ekstrøm CT and Sørensen H. *Introduction to Statistical Data Analysis for the Life Sciences*. CRC Press, 2010.
- [47] Benjamini Y and Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, 1995, 57:289–300.
- [48] Durinck S and Huber W. *biomaRt: Interface to BioMart databases (e.g. Ensembl, COSMIC, Wormbase and Gramene)*. R package version 2.10.0.
- [49] Durinck S and Bullard J. *GenomeGraphs: Plotting genomic information from Ensembl*. R package version 1.14.0.
- [50] Flicek P and et al. . Ensembl 2012. *Nucleic Acids Research*, 2012, 40(Database issue):D84–D90.
- [51] Seal RL, Gordon SM, Lush MJ, et al. gene-names.org: the HGNC resources in 2011. *Nucleic Acids Research*, 2011, (39):514–519.
- [52] Bates D, Maechler M, and Bolker B. *lme4: Linear mixed-effects models using S4 classes*, 2012. R package version 0.999999-0.
- [53] Smyth GK. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 2004, 3(1).
- [54] Yates T. *xmapcore: Core acces to the xmap database (installed separately)*. R package version 1.2.8.
- [55] Stratowa C. *xps: Processing and Analysis of Affymetrix Oligonucleotide Arrays including Exon Arrays, Whole Genome Arrays and Plate Arrays*. R package version 1.10.2.
- [56] Brun R and Rademakers F. ROOT - An Object Oriented Data Analysis Framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1997, 389:81–86.
- [57] Simon R, Lam A, Li MC, et al. Analysis of gene expression data using BRB-Array Tools. *Cancer Informatics*, 2007, 3:11–17.
- [58] Amin SB, Shah PK, Yan A, et al. The dChip survival analysis module for microarray data. *BMC Bioinformatics*, 2007, 12(72).
- [59] Agilent Technologies. Multi-omic Analysis with Agilent’s GeneSpring 11.5 Analysis Platform. Technical Report 5990-7505EN, AgilentTechnologies, Inc., 2011.
- [60] Zhang Z, Lotti F, Dittmar K, et al. SMN deficiency causes tissue-specific perturbations in the repertoire of snRNAs and widespread defects in splicing. *Cell*, 2008, 133:585–600.